A simple collision algorithm for arbitrarily shaped objects in particle‐resolved flow simulation using an immersed boundary method

doi: 10.1002/fld.4826

ORIGINAL ARTICLE

Journal Section

A simple collision algorithm for arbitrarily-shaped objects in particle-resolved flow simulation using an immersed boundary method

Takayuki Nagata[1]* | Mamoru Hosaka[1]† | Shun Takahashi[3]‡ | Ken Shimizu[4]† | Kota Fukuda[5]† | Shigeru Obayashi[6]§

[1]Department of Aerospace Engineering, Tohoku University, Sendai, Miyagi, 980-8579, Japan

[2]Department of Aeronautics and Astronautics, Tokai University, Hiratsuka, Kanagawa, 259-1292, Japan

[3]Department of Prime Mover Engineering, Tokai University, Hiratsuka, Kanagawa, 259-1292, Japan

[4]Institute of Fluid Science, Tohoku University, Sendai, Miyagi, 980-8577, Japan

Correspondence
Department of Aerospace Engineering, Tohoku University, Miyagi, Sendai, 980-8579, Japan
Email: nagata.takayuki@aero.mech.tohoku.ac.jp

Funding information
Japan Society for the Promotion of Science, KAKENHI, Grant Number: 16K18018 and 18K03937

In the present study, we proposed a simple collision algorithm, which can be handled arbitrarily-shaped objects, for flow solvers using the immersed boundary method (IBM) based on the level set and ghost cell methods. The proposed algorithm can handle the collision of the arbitrarily-shaped object with little additional computational costs for the collision calculation because collision detection and calculation are performed using the level set function and image point, which are incorporated into the original IBM solver. The proposed algorithm was implemented on the solid-liquid IBM flow solver and validated by simulations of the flow over an isolated cylinder and sphere. Also, grid and time step size sensitivity on the total energy conservation of objects were investigated in cylinder-cylinder, cylinder-red-blood-cells-shaped (RBC-shaped) objects, sphere-sphere, and sphere-flat plate interaction problems. Through validation, good agreement with previous studies, grid and time step size convergence, and sufficient total energy conservation were confirmed. As a demonstration, the drafting, kissing, and

Abbreviations: DEM, discrete element method; CFD, computational fluid dynamics; IBM, immersed boundary method; DNS, direct numerical simulation; RBC, red blood cell; DKT, drafting-kissing-tumbling; DPN, node par diameter; CFL,Courant-Friedrichs-Lewy.

tumbling (DKT) processes were computed, and it was confirmed that the present result by the proposed method is similar to the previous computations. In addition, particle-laden flow in a channel including obstacles with collision and adhesion phenomena and the interaction of cylinders and wavy-wall were computed. The results of these simulations reveal the capability of solving a flow containing arbitrarily-shaped moving objects with collision phenomena by a simple proposed method. (223/250 words)

KEYWORDS
collision, arbitrarily-shaped particle, immersed boundary, particle-resolved simulation, particle-laden flow, Navier–Stokes

1 | INTRODUCTION

Accurate simulations of particle-laden flows that contain a number of solid particles are important in various fields such as biology, engineering, and so on. The properties of particle-laden flows can be significantly influenced by a solid phase. There are a number of ways to solve particle-laden flows from the point of view of the handling of a dispersed phase, including the point-particle approach and the finite-size particle approach (particle-resolved approach). In the point-particle approach, three types of fluid-particle coupling methods, one-way, two-way, and four-way coupling approaches, are used. The one-way coupling approach considers the effect of the fluid on particles by particle drag models, but particles do not affect a continuum phase. The two-way coupling approach mutually considers the interaction between fluid and particles. In this case, the effects of particles on the fluid are considered to be volume-averaged forces. In addition, the inter-particle interaction is considered in the four-way coupling approach. For example, gy however, the effect of wake vortices released from individual particles cannot be taken into account in the point particle approach.

In the finite-size particle approach, the coupling model obtained by the discrete element method (DEM) [1] and computational fluid dynamics (CFD) [2] has widely been used. This model, i.e., the DEM-CFD model, was proposed by Tsuji et al. [3, 4] In this method, the inter-particle interaction can be taken into account even though the particles are non-circular or non-spherical. However, the DEM-CFD coupling is the one-way coupling approach on the fluid-particle interaction. An immersed boundary method (IBM) is the second option for the finite-size particle approach, which can realize the full-way coupling. In this case, the flow over particles is directly solved using the IBM. This method has been developed and used to simulate an unsteady viscous flow by Udaykumar et al. [5] and Ye et al. [6], and the particulate flow simulations with IBM were conducted by Uhlmann [7]. In addition, the IBM was extended into compressible flow simulations by Ghias et al [8]. In addition, Takahashi et al. [9] and Luo et al. [10] proposed the simplified IBM for compressible viscous flow based on the IBM by Mittal et al [11]. Furthermore, the immersed boundary-Lattice Boltzmann method (IB-LBM) was proposed by Feng and Michaelides [12]. Recent works related to the IBM are mostly confirmed in the applications, such as fluid-motion coupled simulation of a falling plate by Lau et al. [13] and fluid-structure coupled analysis of an elastic object in a compressible flow by Kim et al. [14] Furthermore, progressive achievements related to the IBM are confirmed in studies of turbulent simulation with LBM by Xu et al. [15], application to aeroacoustics problem by Schlanderer et al. [16] and implementation to OpenFOAM by Riahi et al. [17]

The finite-size particle approach has been used in a fundamental study to investigate the effect and behavior of particles. Lucci et al. [18] investigated turbulence modulation effects by spherical particles on decaying isotropic turbulence using the direct numerical simulation (DNS) of the incompressible Navier–Stokes equations with the IBM. Hosaka et al. [19] computed the particle-laden channel flow including an obstacle with collision and adhesion effects using the IBM flow solver and investigated the effect of flow conditions on the adhesion phenomena behind an obstacle under low-Reynolds number conditions. Mehrabadi et al. [20] examined the effects of particle configurations on the mean drag of clustered particles using the IBM and developed a gas-solid drag law for clustered particles. Zhang et al. [21] investigated the effect of collisions on particle behavior in turbulent flows through a straight square duct by DNS with the DEM. Also, Mizuno et al. [22] computed the collision of the particle cluster and wall, and they investigated the effect of the wake generated by particles on the distribution of collisions. In addition, particle-resolved DNS has been applied to compressible flow simulations. Schneiders et al. [23] developed an efficient cut-cell method for rigid bodies interacting with the viscous compressible flow, and they computed particle-laden isotropic turbulence. Das et al. [24] applied a Cartesian grid-based sharp interface method for the compressible particle-laden flow, and they demonstrated the interaction of a particle cloud and planar shock.

A Collision between particles or particle-wall should be taken into account in particle-laden flows, particularly when the particle volume fraction is high. The particles are typically modeled as cylindrical or spherical particles in analyses of two- or three-dimensional particle-laden flows. In this case, hard-sphere collision models [1, 25, 26] and the other advanced hard-sphere (e.g., [27]) or soft-sphere (e.g., [28]) models are typically used to compute collision process. Recently, the particle-resolved approach has also been used for simulation of particle-laden flows with collisions. Also, interaction of the interface of the liquid and solid phase is treated by the IBM (e.g., [29]). In such a case, the contact line is the arbitrary shape and the normal vectors of the contact line of the liquid phase and solid phase should be equal.

In the case of the particle-resolved approach, overlap of the particles or particle and wall should be avoided even if the collision phenomena are not dominant in the system. The collision for the simple geometry such as spheres or cylinders, it can simply compute the collision by using the radius and the position of each object. However, that kind of simple treatment is not available for arbitrarily-shaped objects. Also, the effect of collisions has a large impact on the behavior of arbitrarily-shaped particles because of irregular bouncing, the effect of particle rotation, and so on. In the DEM-CFD and IBM based simulations, the collision model proposed by Glowinski et al. [30, 31], which is Lagrange-multiplier based fictitious-domain method, is widely used in particulate flow simulations. [7, 12, 32, 33] In addition, Ardekani et al. [34] examined the sedimentation of spheroidal particles using the IBM with a collision algorithm that can handle spheroidal particles based on the soft-sphere model by Costa et al. [28] In their study, the spheroidal particles were approximated as spherical particles with the same mass as the whole particle and with a radius corresponding to the local curvature at the contact point.

The DEM-CFD coupling is widely used for the flow simulation including arbitrarily-shaped objects with collisions. In DEM simulations, arbitrarily shaped objects are expressed by the mass point, and collision detection and reactions are treated as an inter-particle collision. However, the collision detection requires the computational cost and it rapidly increases as the number of particles increases. In practice, the contact detection has a considerable cost in the DEM for a system with a large number of particles. Therefore, several contact detection algorithms have been proposed to reduce the computational cost [35, 36, 37]. Zhang et al. [38] proposed an efficient collision algorithm for spheroidal particles for the DEM-CFD (LBM) flow solver. They reduced the computational cost for collision detection by using a lattice grid for the lattice Boltzmann simulation to detect the contact of the object. However, collision detection algorithms for the DEM require a large amount of additional computation because object boundaries in the IBM flow solver are defined in a Eulerian form by level set functions. Accordingly, we proposed a simple and efficient collision algorithm for the IBM flow solver that can handle arbitrarily-shaped objects. This algorithm requires only little additional computations

for contact detection because the present algorithm uses the image point and level set function, which are essential components of the IBM flow solver.

2 | METHODOLOGIES

2.1 | Basic Equations

The fluid was assumed to be a Newtonian fluid and objects are assumed to be rigid bodies. The incompressible Navier–Stokes equations

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{1}$$

and the equation of continuity

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2}$$

were used as the governing equations for the fluid. Here, i and j are the indices for the Einstein summation convention, u is the fluid velocity, p is the static pressure, x is the Cartesian coordinate, t is time, and Re is the Reynolds number based on freestream quantities and the representative length. The governing equations were non-dimensionalized by freestream quantities and the representative length and were discretized by the finite difference method. The fractional step method proposed (e.g., [39]) was used in the present study for the pressure-velocity coupling. In this method, the temporal velocity u^* is introduced and the Navier–Stokes equations are split as follows:

$$\frac{u_i^* - u_i^n}{\Delta t} + \frac{\partial (u_i u_j)^n}{\partial x_j} = \frac{1}{Re}\frac{\partial^2 u_i^n}{\partial x_j \partial x_j}, \tag{3}$$

$$\frac{u_i^{n+1} - u_i^*}{\Delta t} = -\frac{\partial p^{n+1}}{\partial x_i}, \tag{4}$$

where the first-order Euler explicit method was used for the time integration, and superscripts n and $n+1$ indicate the current step and next step in the time direction. Also, the second-order central difference method By taking the divergence of eq. 4, the following equation can be obtained:

$$-\frac{\partial^2 p^{n+1}}{\partial x_j \partial x_j} = \left(\frac{\partial u_j^{n+1}}{\partial x_j} - \frac{\partial u_j^*}{\partial x_j}\right)\frac{1}{\Delta t}. \tag{5}$$

Here, eq. 2 will be satisfied at the $n+1$ step so that the pressure Poisson equation of eq. 5 is derived as follows:

$$\frac{\partial^2 p^{n+1}}{\partial x_j \partial x_j} = \frac{\partial u_j^*}{\partial x_j}\frac{1}{\Delta t}. \tag{6}$$

Both sides of the pressure Poisson equation was discretized by the second-order central difference method and solved by the successive over-relaxation method (see [40]). The convection term in eq. 3 was evaluated by the hybrid scheme of the second-order conservative central difference method and the first-order upwind method shown below:

$$\begin{aligned}\left(\frac{\partial(u_i u_j)}{\partial x_j}\right)_I &= (1-\alpha)\left[\frac{(u_i u_j)_{I+1/2} - (u_i u_j)_{I-1/2}}{2\Delta x_j}\right] \\ &+ \alpha(u_j)_I\left[\frac{(u_i)_{I+1} - (u_i)_{I-1}}{2\Delta x_j} + \frac{(u_i)_{I+1} - 2(u_i)_I + (u_i)_{I-1}}{2\Delta x_j}\right]\end{aligned} \tag{7}$$

where the subscript I denotes the quantity at the I th grid point and its sweep direction is the same as the direction of the spatial differential. In addition, α is a parameter to determine the ratio of the second-order conservative central difference method and the first-order upwind method, and its value was set to be $\alpha = 0.05$ in the present study. The viscous term was evaluated by the second-order central difference method.

The motion of objects is described by Newton–Euler equations,

$$m\frac{\mathrm{d}u_{\mathrm{obj}\,j}}{\mathrm{d}t} = \int_{\Gamma_S} \sigma_{ij} n_i \mathrm{d}S + G_j, \tag{8}$$

$$\frac{\mathrm{d}(I\omega)_i}{\mathrm{d}t} = \int_{\Gamma_S} \epsilon_{ijk} r_j \sigma_{kl} n_l \mathrm{d}S + H_i. \tag{9}$$

Equation 8 describes a translational motion of the objects where m and u_{obj} indicate the mass and velocity of the object, and σ, n, $\mathrm{d}S$, and G denote the stress tensor of a Newtonian fluid, the unit vector in the normal outward direction of the surface element, the micro-area element, and the external force. Equation 9 describes a rotational motion of the objects where I, ω, r, ϵ, and H indicate the moment of inertia of objects, the angular velocities of objects, the relative position vector from the centroid, the third-rank tensor of Eddington's epsilon, and external moment. Since the mesh-based scheme proposed by Nonomura and Onishi [41] was used as a hydrodynamic force evaluation method, the micro-area element $\mathrm{d}S$ in the two- and three-dimensional cases for the present solver was equal to Δx and Δx^2, respectively. The stress tensor of a Newtonian fluid is given by

$$\sigma_{ij} = -P\delta_{ij} + \frac{1}{Re}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \tag{10}$$

where δ is the Kronecker delta. The time integration of Newton–Euler equations was performed by the first-order Euler explicit method.

2.2 | Immersed Boundary Method

Object boundaries were treated by the IBM based on the level set and ghost cell methods. In the present study, we used the method originally proposed by Takahashi et al. [9] The IBM can reduce the computational cost of grid regeneration in a problem involving multiple moving objects. In the present study, the computational grid was a fixed and equally spaced Cartesian mesh. Figure 1 shows a schematic diagram of the current IBM. The grid points are classified into fluid, ghost, and object nodes by the following criteria:

$$\begin{array}{ll} \phi > 0, & \text{Fluid node} \\ -l_{\text{probe}} < \phi < 0, & \text{Ghost node} \\ \phi < -l_{\text{probe}}, & \text{Object node} \end{array} \tag{11}$$

where ϕ is the level set function that represents the signed minimum distance between the object surface and each node, and l_{probe} is the probe length. In the present study, in order to avoid a recursive reference in the ghost node calculations, the probe length l_{probe} was set to be $1.45\Delta x$ or $1.75\Delta x$ in the two- or three-dimensional computations. The boundary conditions on the object surface were imposed on the ghost nodes using quantities at the image points and the level set function. Here, the image point indicates the tip of a probe that is extended toward the exterior of objects from ghost nodes normal to the object surface. The pressure and velocity at the image points were calculated by the bilinear or trilinear interpolation using the surrounding nodes of the image point. The pressure at the ghost nodes is determined by the zeroth-order extrapolation from the image point. The velocity at the ghost node u_{GN} was determined by the linear extrapolation so as to satisfy the no-slip condition at the object surface.

$$u_{\text{GN}\,i} = u_{\text{IP}\,i} - \frac{l_{\text{probe}} + |\phi_{\text{GC}}|}{l_{\text{probe}}}(u_{\text{IP}\,i} - u_{\text{surf}\,i}) \tag{12}$$

$$P_{\text{GN}} = P_{\text{IP}} \tag{13}$$

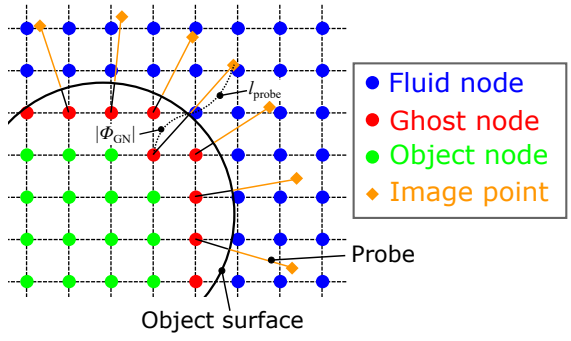

FIGURE 1 Schematic diagram of the IBM.

2.3 | Collision Algorithm for an Arbitrarily-Shaped Object in IBM Flow Solver

A simple collision algorithm for arbitrarily-shaped objects is explained in this section. The computational cost for contact detection becomes larger as the number of particles increases, and thus the contact detection has considerable computational cost in the DEM with a large number of particles. Accordingly, we proposed a simple and efficient collision algorithm for a flow solver using the IBM. The computation of collision between arbitrarily-shaped objects has two difficulties from a computational point of view: search for the point of contact and calculation of the local curvature. In our algorithm, the image point that is essential in the IBM is used to detect the contact and contact point. In addition, the local curvature at the contact point can be calculated from the level set function, which is a fundamental technique in the current IBM. The present method can remove the computational cost for collision detection because collision detection can conduct simultaneously in the ghost nodes computations which are incorporated into the original IBM flow solver. In the original IBM solver, an exception handling on the ghost node calculations is required when the image point refers to the ghost nodes belonging to the other object. Our algorithm conducts the collision detection by the information of the exception handling of the ghost nodes so that collision calculation can be conducted with no additional cost for contact detection and small additional cost for local curvature at the contact point, collision angle, collision force, and so on. As a result, only little additional computational costs is needed to treat the collision of arbitrarily-shaped objects. In the present study, the objects were assumed as rigid bodies to simplify the problem. A schematic diagram of the contact detection is shown in Figure 2. The procedure from collision detection to the calculation of the collision force is listed below:

1. Extend probes from each ghost node outward normal to the object surface and create image points (Figure 2A). In our algorithm, the lengths of the probes for the ghost cell method and collision determination are the same to reduce the computational cost. For this assumption, computations of the quantities at ghost nodes and collision detection can be performed at the same time, and thus the computation of the position of the image point and collision detection are completed before the computation of collisions. The positions of the image point x_{IP} are calculated as follows:

$$x_{\mathrm{IP}\,i} = x_{\mathrm{GC}\,i} + |\phi| n_i + l_{\mathrm{probe}}. \tag{14}$$

2. Find image points neighboring ghost nodes and determine the contact point nodes (this process can be combined with the calculation of quantities at image points) (Figure 2B).

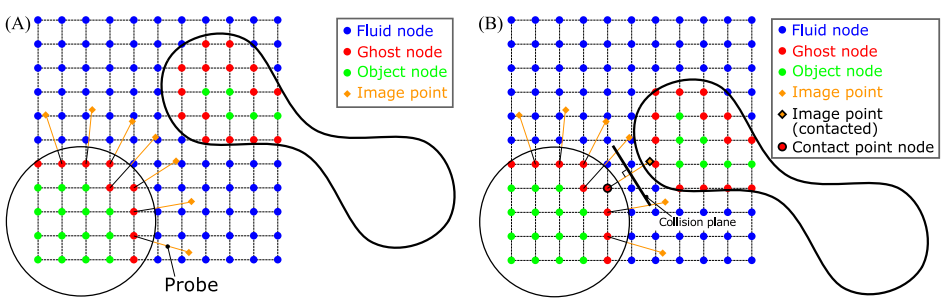

FIGURE 2 Schematic diagram of the contact detection process: (A) before collision; (B) during collision.

Here, the present algorithm does not include the collision time prediction, but collision detection is conducted by using image points. Hence, there is no overlapping of the object if the relative moving distance between contact points of each collided object within Δt is less than l_{probe}. Instead of that, however, the small clearance ($< l_{\mathrm{probe}}$) between colliding objects exists, and it might be causing the difference with the exact solution of the collision phenomena. It should be noted that the error due to the small clearance is reduced by decreasing the grid size because l_{probe} depends on the grid size in the present method to reduce the computational cost. The influence of the grid size will be discussed in Section 3.2. The collision time prediction or additional process to independent from the grid size limitation is required for computations with large Δt and small l_{probe} cases.

3. Find a pair of colliding objects and calculate the $x-$ and $y-$components of the unit normal vector n at the contact surface by the level set function around the contact point node as follows:

$$n_i = \frac{\partial \phi}{\partial x_i} \frac{1}{\sqrt{\left(\frac{\partial \phi}{\partial x_j}\right)^2}} \tag{15}$$

4. Calculate the impulsive stress acting on the contact point of object A due to collision with object B based on the equation of the elastic collision for two- or three-dimensional rigid bodies (see [42]) as follows:

$$\boldsymbol{\sigma}_{\mathrm{colli\ AB}} = \frac{(1+\epsilon)\left[\left(\mathbf{u}_{\mathrm{obj\ B}} - \mathbf{u}_{\mathrm{obj\ A}}\right) + \{(\boldsymbol{\omega}_{\mathrm{B}} \times \mathbf{r}_{\mathrm{B}}) - (\boldsymbol{\omega}_{\mathrm{A}} \times \mathbf{r}_{\mathrm{A}})\}\right] \cdot \mathbf{n}_{\mathrm{BA}}}{m_{\mathrm{A}}^{-1} + m_{\mathrm{B}}^{-1} + \mathbf{n}_{\mathrm{AB}}\left[\left\{I_{\mathrm{A}}^{-1}\left(\mathbf{r}_{\mathrm{A}} \times \mathbf{n}_{\mathrm{AB}}\right)\right\} \times \mathbf{r}_{\mathrm{A}} + \left\{I_{\mathrm{B}}^{-1}\left(\mathbf{r}_{\mathrm{B}} \times \mathbf{n}_{\mathrm{BA}}\right)\right\} \times \mathbf{r}_{\mathrm{B}}\right]} \frac{1}{\mathrm{d}t\mathrm{d}S} \tag{16}$$

where subscripts A and B indicate objects A and B, respectively, ϵ is the coefficient of restitution, and $\mathbf{n}_{AB}$ is the normal vector from object A to object B. The calculated impulsive stress is mapped to the node at the contact point. In the present study, the equation of the elastic collision was used to calculate the collision (interaction force between objects or an object and wall) force, but it can be changed that the estimation method of the collision force depending on the problem settings.

5. Calculate the force and moment acting on each object by integrating the hydrodynamic stress $\boldsymbol{\sigma}_{\mathbf{fluid}}$ and additional stress such as the stress due to collision $\boldsymbol{\sigma}_{\mathrm{coll}}$ and adhesion $\boldsymbol{\sigma}_{\mathrm{adh}}$ at the ghost nodes.

$$\mathbf{F} = \int_{\Gamma_S} \left(\boldsymbol{\sigma}_{\mathrm{fluid}} + \boldsymbol{\sigma}_{\mathrm{adh}} + \boldsymbol{\sigma}_{\mathrm{colli}} + \cdots\right) \mathbf{n}\mathrm{d}S + \mathbf{G}, \tag{17}$$

$$\mathbf{M} = \int_{\Gamma_S} \mathbf{r} \times \left(\boldsymbol{\sigma}_{\mathrm{fluid}} + \boldsymbol{\sigma}_{\mathrm{adh}} + \boldsymbol{\sigma}_{\mathrm{colli}} + \cdots\right) \mathbf{n}\mathrm{d}S + \mathbf{H}, \tag{18}$$

Here, in the present result shown in Section 4.2, the stress due to adhesion was estimated by the adhesion model for platelet adhesion proposed by Tomita et al. [43], and the stress due to adhesion does not consider in the other simulations. In addition, other forces such as the friction force due to collision and so on can be taken into account if users include models and impose the stress at the ghost node of the contact point. The stresses are integrated at the surface of the object and motion of the objects obeys the Newton–Euler equations.

6. Solve the Newton—Euler equations and calculate the velocity and angular velocity of the object.

$$\mathbf{u}_{\mathrm{obj}} = \frac{1}{m}\int \mathbf{F}\mathrm{d}t, \tag{19}$$

$$\boldsymbol{\omega} = \frac{1}{I}\int \mathbf{M}\mathrm{d}t, \tag{20}$$

Steps 1 and 2 have already done to determine the velocity and pressure at the ghost nodes. For this reason, this algorithm requires only little additional computational costs for the treatment of collisions of objects, and the algorithm has a high affinity for the IBM. However, it appears to be that the calculation accuracy of the collision phenomenon using the proposed method would be affected by the grid width instead of simplicity. In addition, it is difficult to treat the collision of objects which have sharp part accurately due to computational cost, because sufficient grid point is necessary to compute the curvature of the surface at the contact point accurately. The required grid resolution will be discussed in Section 3.2. Figure 3 shows a flowchart of the computational procedure. Computations for the fluid and object are shown in the blue and red boxes, respectively. In the case of computation including moving objects, the process of moving objects is used. This process contains computation of the Newton–Euler equations, the collision detection, and other processes related to moving objects such as regeneration of the level set function and so on.

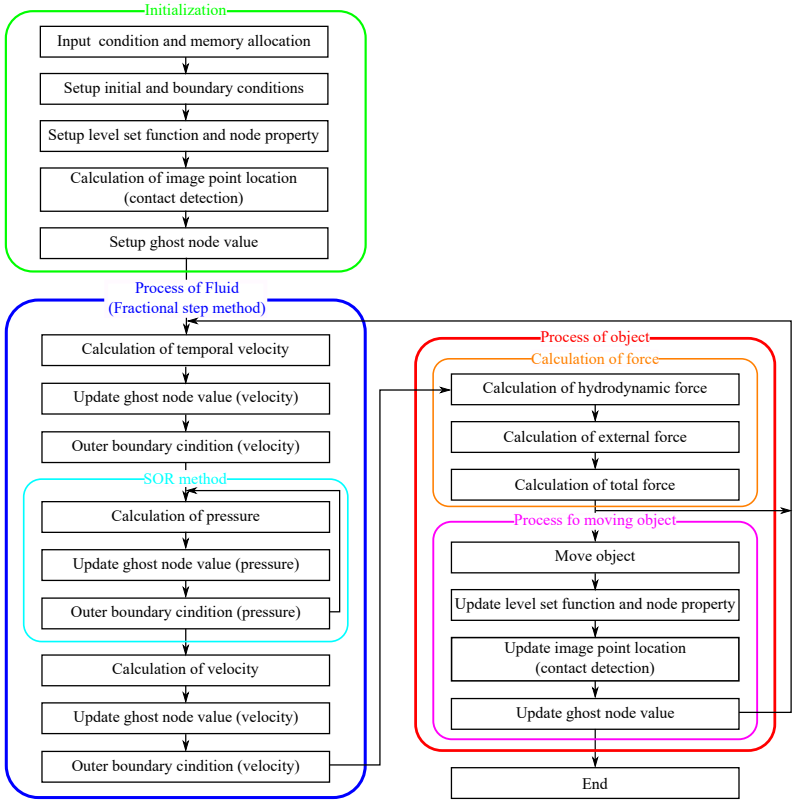

FIGURE 3 Flowchart of flow solver.

3 | NUMERICAL TESTS

3.1 | Flow Past a Circular and Sphere

The flows past a circular and sphere were calculated. For the two-dimensional case, the results were compared to the results of previous studies by Zhang et al. [44], Mittal et al. [11], and Takahashi et al. [9]. The Reynolds number based on freestream quantities and the diameter of the cylinder was set to be 300, and the grid size was set to be $0.1D, 0.05D, 0.025D, 0.0125D, 0.00833D$, or $0.00625D$ (where the number of nodes per diameter (NPD) was 10, 20, 40, 80, 120, or 160). The computations were conducted at a Courant-Friedrichs-Lewy (CFL) number of 0.1. The computational domain was $0 \leq x \leq 30D$ and $0 \leq y \leq 20D$, and the position of the center of the cylinder was at $(x, y) = (10D, 10D)$. The inflow (the velocity is fixed at the freestream velocity, and the pressure is a zeroth-order extrapolation from one point inside of the boundary) and outflow (the velocity is a zeroth-order extrapolation from one point inside of the boundary, and the pressure is fixed at the freestream value) conditions were imposed at the $-x$ and $+x$ boundaries, and the slip wall condition was imposed at the $-y$ and $+y$ boundaries.

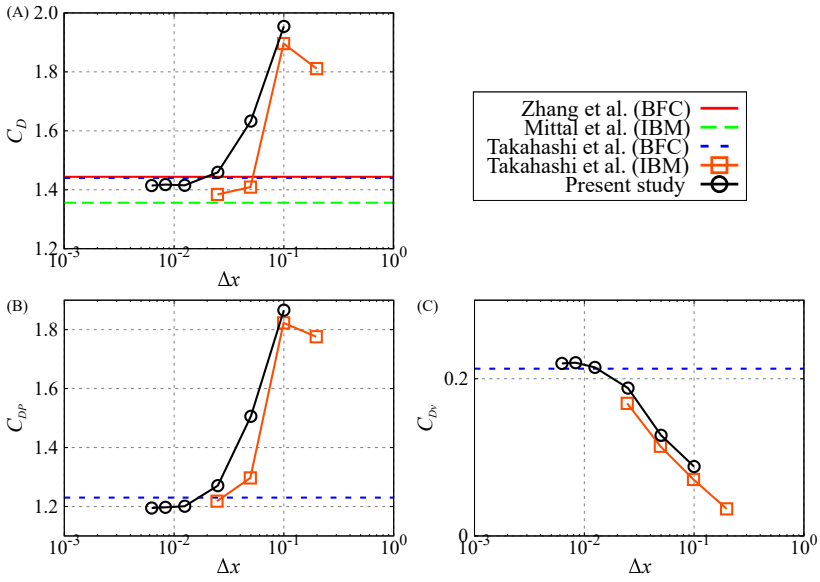

FIGURE 4 Influence of the grid size on the drag coefficients of the circular cylinder ($Re = 300$): (A) total drag coefficient, (B) pressure drag coefficient, and (C) viscous drag coefficient.

Figure 4 shows the relationship between grid size and the time-averaged drag coefficient for a Reynolds number of 300. The calculations were carried out for a duration longer than 10 vortex-shedding periods. The result reported by Zhang et al. [44] was obtained under the incompressible flow on a boundary-fitted grid, and the result reported by Mittal et al. [11] was obtained under an incompressible flow using the IBM. The result reported by Takahashi et al. [9] was obtained under a compressible flow of the Mach number of 0.3 using a boundary-fitted grid and the IBM solver. Their results are under the compressible flow, but they provided the drag coefficient for each component. In addition, there is no large impact of the compressibility on the drag coefficient at the Mach number of 0.3. Here, all of the results that were compared to the present study were obtained by a fully two-dimensional computation. Figure 4 illustrates

that the pressure and viscous drag coefficients were overestimated and underestimated, respectively, at the coarse grid. The difference between the present and previous results decreases as the grid resolution increases.

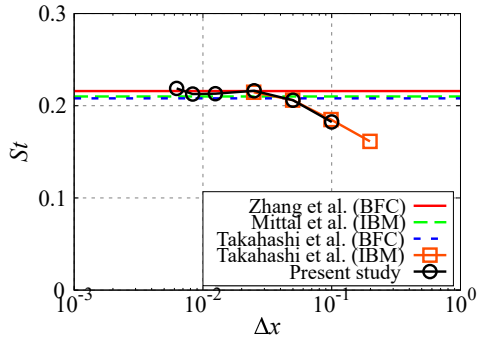

FIGURE 5 Influence the grid size on the Strouhal number of the circular cylinder ($Re = 300$).

Figures 5 and 6 show the relationship between the grid size and the Strouhal number and root-mean-square (r.m.s.) amplitude of the lift coefficient, respectively. The Strouhal number and r.m.s. of the lift coefficient of the present study show good agreement with the previous computations.

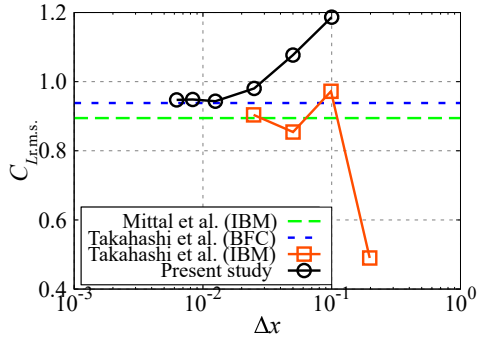

FIGURE 6 Influence of the grid size on $C_{L\ \mathrm{r.m.s.}}$ of the circular cylinder ($Re = 300$).

In the three-dimensional case, the drag coefficient of a sphere was compared to the previous studies [11, 45, 46, 47, 48]. The drag model by Clift and Gauvin [45] is known as the standard drag curve of a sphere. The resutl by Johnson and Patel [46] is the DNS result on a boundary-fitted grid, the results by Luo et al. [10] and Mittal et al. [11] are the DNS results by the IBM code. In addition, the result by Nagata et al. [48] is the DNS result on a boundary-fitted grid at the Mach number of 0.3, and they showed pressure and viscous drag coefficients separately. The computation of the present study was carried out at the Reynolds numbers of 100, 250, 300, and 500. The grid size was $\Delta x = 0.125D, 0.1D, 0.05D, 0.025D$, or $0.0125D$ (where NPD was 8, 10, 20, 40, or 80). The size of the computational domain was $0 \le x \le 30D$, $0 \le y \le 20D$, and $0 \le z \le 20D$, and the sphere was placed at $(x, y, z) = (8.5D, 10D, 10D)$. Figure 7 illustrates that the drag coefficient of the present study shows good agreement with previous studies. However, the difference between the previous results and the present study becomes large for $Re \ge 250$ with small NPD. The influence of the grid size on the drag coefficient at the Reynolds number of 300 is shown in Figure 8. The present results

overestimate the total and pressure drag coefficients and underestimates the viscous drag coefficient at the coarse grid, but grid convergence has been confirmed for each component as grid size decreases.

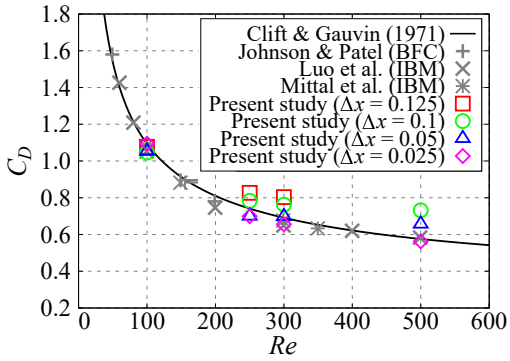

FIGURE 7 Comparison of the drag coefficient of the sphere at various Reynolds numbers.

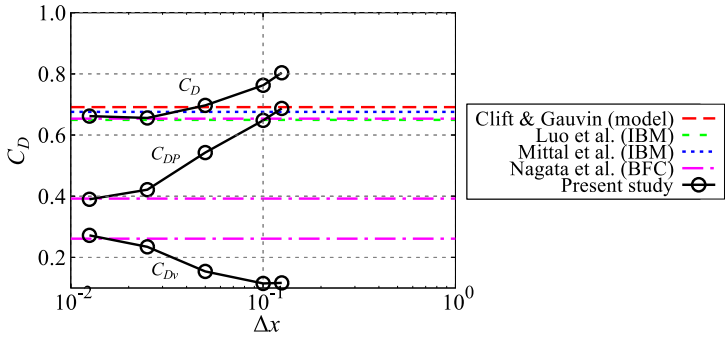

FIGURE 8 Influence of the grid size on the drag coefficient of the sphere ($Re = 300$).

3.2 | Collision of Objects without Solid-Fluid Interaction

Figure 9 shows the influences of the grid and time-step sizes on the relative error in the total energy conservation of an object before and after collision. The total energy of the object was defined as follows:

$$\mathbf{E} = \frac{1}{2}\left(m|\mathbf{u}_{\mathrm{obj}}^2| + I\omega^2\right) \tag{21}$$

The interactions between cylinder-cylinder and sphere-sphere without solid-fluid interactions were computed in order to investigate the kinetic energy conservation regarding the collision. The diameter and density of the objects was unity. In addition, the coefficient of restitution was unity so that the total energy of the object should be conserved. However, since collision is partially treated based on nodes and image points in the proposed method, the total energy of an object is not strictly conserved except when the collision point matches the grid line.

The computational domain was $0 \le x \le 10D$ and $0 \le y \le 10D$ (and $0 \le z \le 10D$ in the sphere case). The outflow

boundary condition was imposed at all outer boundaries. In this test, a stationary cylinder or sphere was placed in the center of the computational domain of $(x_{obj}, y_{obj}) = (5.0D, 5.0D)$ (and $z_{obj} = 5.0D$ in the sphere case). The initial position of a moving cylinder or sphere was at $(x_{obj}, y_{obj}) = (3.0D, 5.01D)$ (and $z_{obj} = 5.0D$ in the sphere case), and the initial velocity was $(u_{obj}, v_{obj}, w_{obj}) = (1.0, 0, 0)$. Since the total energy before and after the collision was fully conserved when the collision point matches the grid lines, there is a small offset on the object position in the y-direction to confirm the grid and time-step size dependencies on the kinetic energy conservation. In addition, the computation was conducted five times by changing the initial position in the y-direction of the moving object by $y_{obj} = 0.1D$, and the post-collision kinetic energy E_{post} was averaged. The grid size was set to be NPD = 10, 20, 40, 80, 120, or 160 for the cylinder case and was set to be NPD = 5, 8, 10, 20, or 40 for the sphere case. The CFL number based on grid size and initial velocity of the moving object was set to be CFL = 0.5, 0.3, 0.2, 0.1, or 0.05 for the cylinder case and was set to be CFL = 0.2, 0.1, or 0.05 for the sphere case. Figure 9(A) illustrates that the total energy conservation improved as the grid resolution increases. In the case of the cylinder-cylinder interaction, the error for NPD = 10 is less than or equal to 2.0% and decreases to less than or equal to 0.63% and 0.27% at NPD = 20 and 40, respectively. Also, Figure 9(B) illustrates that the effect of CFL on the kinetic energy conservation is weak. The effects of NPD and CFL on total energy conservation in the sphere-sphere case are similar to the cylinder case.

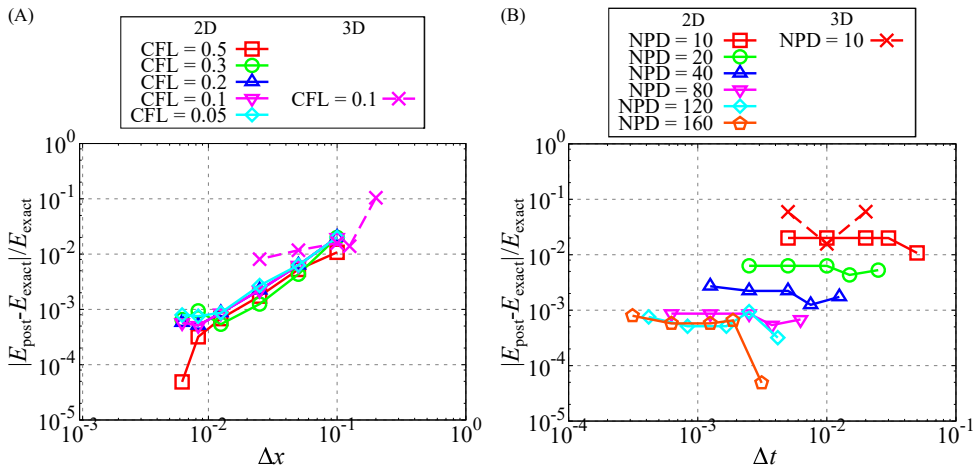

FIGURE 9 Influence of (A) Grid size and (B) time step size on total energy conservation of the collision process.

Figure 10 shows the distribution of the level set function during cylinder-RBC-shaped object interactions as an example. The size of the computational domain, the inner and outer boundary conditions, and the coefficient of restitution were the same as for the cylinder-cylinder interaction problem. The initial position and velocity of the cylinder were $(x_{cylinder}, y_{cylinder}) = (5.125D, 5.5D)$ and $(u_{cylinder}, v_{cylinder}) = (1.0, 0)$, respectively, and the initial position and velocity of the RBC-shaped object were $(x_{RBC}, y_{RBC}) = (6.0D, 5.0D)$ and $(u_{RBC}, v_{RBC}) = (0, 0)$. In the present study, the shape of the RBC was set according to Chee et al. [49]. The diameter of the cylinder and the major axis of the RBC was equal to unity. In addition, the CFL number and NPD were set to be 0.1 and 80, respectively. Figure 10 illustrates that the RBC-shaped object rotates in the post-collision state, and the total energy conservation was 0.012%. Hence, the total energy was sufficiently conserved even in the case including the object rotation. The present algorithm conducts the collision calculation based on image points, ghost nodes, and the level set function so that the process of the collision calculation is regardless of the object shape. The surface stress due to collision is calculated by Eq. 16, and the stress

is imposed on the ghost nodes at the contact point. Eventually, stresses due to the collision and other factors such as fluid, adhesion, and so on are integrated on the object surface, and the equations of motion for the object are solved to determine the object motion.

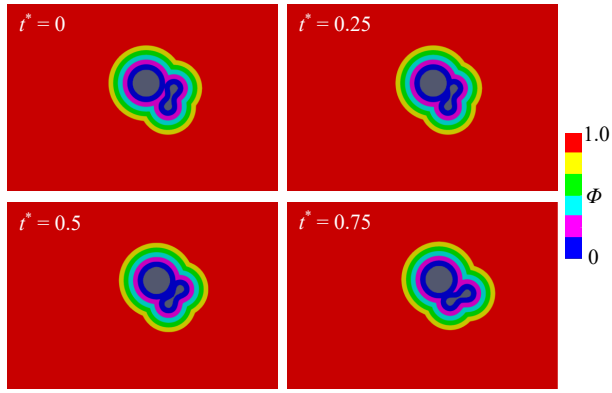

FIGURE 10 Distribution of the level set function for cylinder-RBC interaction.

3.3 | Collision of a Sphere and a Flat Plate with Solid-Liquid Interaction

The collision of a sphere and a flat plate with solid-liquid interaction was calculated and compared to the experimental result by Thompson et al. [50] In their experiment, a brass sphere of 19.02 mm in diameter was attached to a fine thermally fused twisted stretch-resistant thread. By a stepper motor, the sphere was allowed to be lowered through the water at a specified constant velocity. In the present computations, the Reynolds number based on the falling velocity, the diameter of the sphere, and the fluid quantities was set to be 500 as the same with the experiment of Thompson et al. [50] The size of the computational domain was set to be $(x, y, z) = (15D, 15D, 10D)$, and the initial distance between the center of the sphere and surface of the flat plate was $5.5D$ as the same with the experiment. The flat plate is placed on the bottom boundary of the computational domain and expressed by the immersed boundary as well as the sphere. The grid resolution and the CFL number were set to be NPD $= 20$ and CFL $= 0.1$, respectively. In addition, in the experiment of Thompson et al. [50], the sphere was stopped at the surface of the flat plate so that the collided sphere does not bounce. Therefore, the coefficient of restitution ϵ was set to be zero in the present computation. Figure 11 shows the distribution of the vorticity in the $y-$direction, which was normalized by the diameter of the sphere and the initial velocity magnitude of the sphere. The non-dimensional time t^* was normalized by the diameter of the sphere and the initial velocity magnitude of the sphere. Before the impact, the wake develops similarly to the flow in the isolated sphere, and $t^* = 0$ corresponds to the time of the impact. Owing to the short falling distance, the wake of the sphere is a steady-axisymmetric wake even though $Re = 500$. At $t^* = 0$, the sphere collides with a flat plate and stopped impulsively. At $t^* = 1$ and 2, the wake catches up with the sphere and the secondary vortex is generated due to the induced flow generated by the primary vortex. After $t^* = 3$, the primary and secondary vortices arrive at the flat plate and leave from the sphere in the horizontal direction. Our computational results qualitatively agree with experimental flow visualization by Thompson et al. [50]. Since the length of prob l_{probe} for collision detection, however, there is a small clearance between the sphere and flat plate in Figure 11. This clearance can be reduced by decreasing Δx or l_{probe}. If the additional computational cost to avoid recursive reference for ghost node calculation is acceptable, l_{probe} can be reduced at the constant Δx.

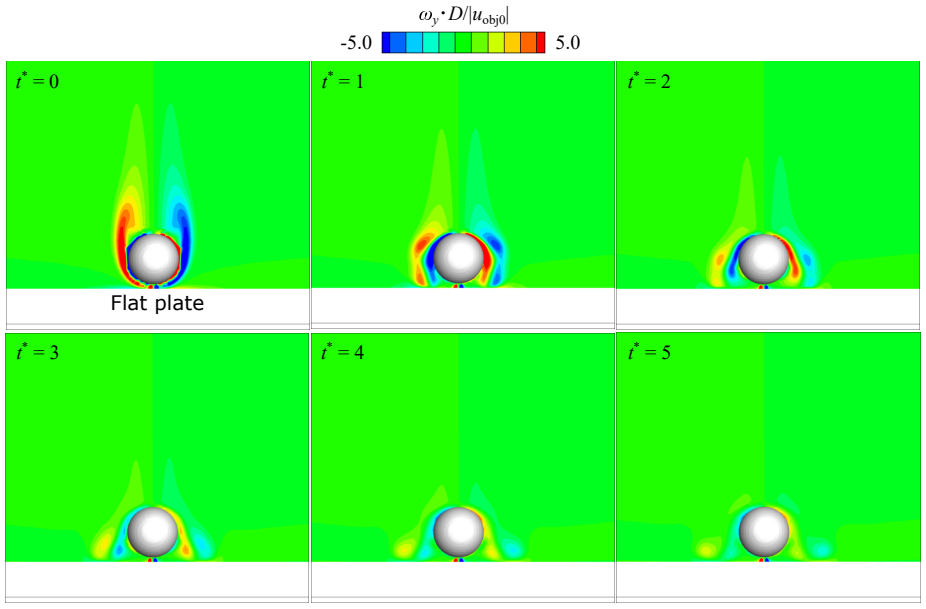

FIGURE 11 Snapshot of the vorticity distribution of a sphere-flat plate collision in the $x - z$ plane at $y = 7.5$ ($e = 0$).

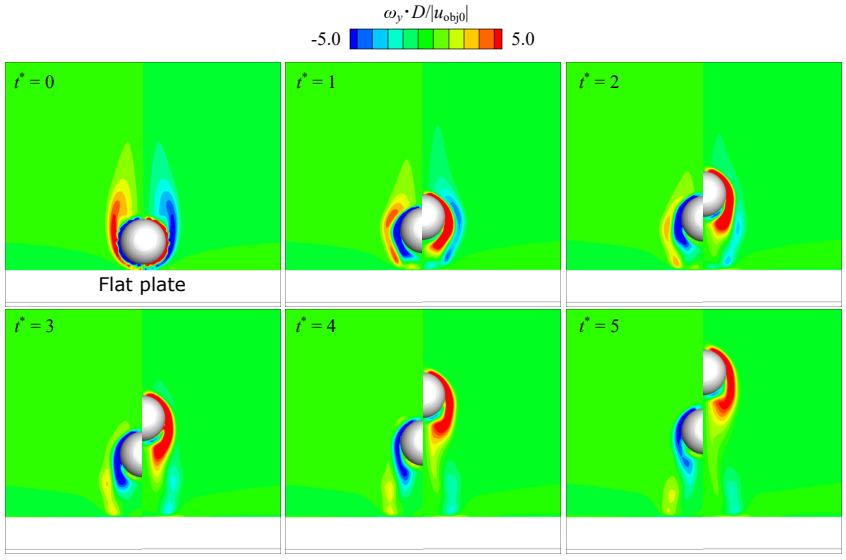

FIGURE 12 Vorticity distributions of the sphere-flat plate collision for different values of e. The left- and right-hand parts for each time show the results for $e = 0.5$ and 1.0, respectively. The visualization plane is $x - z$ plane at $y = 7.5D$.

Figure 12 shows the instantaneous vorticity distributions for different coefficients of restitution. The coefficient of restitution for the left and right halves of each frame is 0.5 and 1.0, respectively. The development of the vortex

structures around the sphere and on the flat plate is similar for $\epsilon = 0.5$ and 1.0. The object velocity after bounce for $\epsilon = 0.5$ is half of the case of $\epsilon = 1.0$. The primary vortices in the vicinity of the flat plate for the bouncing case do not spread in the horizontal direction compared with $\epsilon = 0$.

4 | DEMONSTRATIONS

4.1 | Drafting-Kissing-Tumbling Process

A free-falling problem of cylindrical objects placed in-line and pulled by gravity in the same direction was computed. In this situation, the cylinders undergo a characteristic behavior called drafting-kissing-tumbling (DKT) [51]. The calculation results were compared to those of previous numerical studies [7, 30, 52]. The density ratio of fluid and cylinders was set to be $\rho_c/\rho_f = 1.5$. The Reynolds number based on the fluid density, the viscosity coefficient μ, the diameter of the cylinder D, and the estimated terminal velocity U_t was set to be 347. Here, the estimated terminal velocity was calculated as follows [52]

$$U_t = \sqrt{\frac{\pi D}{2}\left(\frac{\rho_c - \rho_f}{\rho_f} g\right)}, \tag{22}$$

where g is the acceleration of gravity. The computational domain was $0 \leq x \leq 10D$ and $-40D \leq x \leq 0$, and the grid resolution was set to be NPD = 40. The pair of cylinders were placed in-line in at $x = 5D$ (center of the computational domain in the $x-$direction). The trailing and leading cylinders were placed at $y = -4D$ and $-6D$, respectively. It should be noted that the position in the $x-$direction of the leading and trailing cylinder was $\pm D/250$ off-center. In the present computation, the gravity and buoyancy forces act at the centroid of each object, and the coefficient of restitution was set to be unity for both cylinders. The outflow boundary condition was imposed for all outer boundaries. The velocities of the cylinders and fluid were zero in every direction at the initial state.

Figure 13 shows the snapshots of the vorticity distribution during the DKT process, and the dynamic interaction between the two cylinders can be observed. Initially, both cylinders begin falling by the effect of gravity with the same acceleration. The trailing cylinder is attracted to the wake of the leading cylinder, and its falling velocity increases (drafting phase). Eventually, the cylinders contact each other (kissing phase) and fall in tandem as an elongated object in the gravitational direction. However, such a configuration is unstable. As a result, the position in the $x-$direction begins to differ from the initial position, and the elongated object rotates so that the trailing cylinder overtakes the leading cylinder (tumbling phase). The quantitative comparisons are shown in Figure 14. These figures show the time histories of the position and velocity of the centroid of the cylinders in the $x-$ and $y-$directions. The positions and velocities were nondimensionalized by the diameter of the cylinder and the estimated terminal velocity. Figure 13 illustrates that the time histories of the position and velocity of the center of the cylinders of the present result exhibit quantitatively agree with the previous studies. However, the difference between each study becomes large at a later time. Koblitz et al. [52] pointed out the difference between their results and Uhlmann [7] during the initial contact and subsequent kissing phase is caused by the difference in the collision model. It should be noted that the DKT problem is very sensitive so that a relatively large difference in subsequent kissing phase appears due to a difference in the initial stage even if a very small difference.

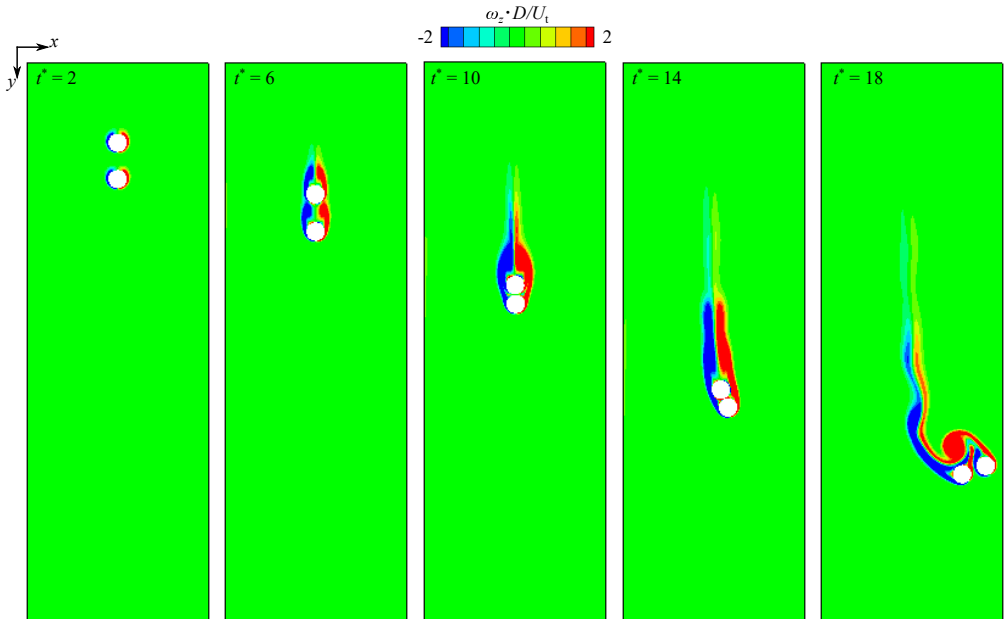

FIGURE 13 Snapshot of vorticity distribution for the DKT process.

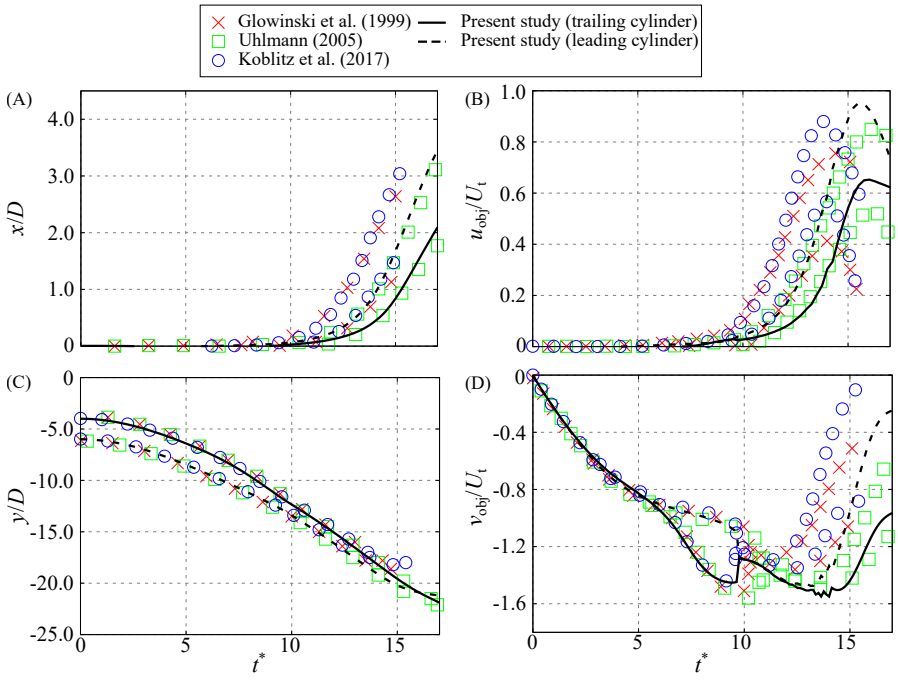

FIGURE 14 Time histories of (A) x^*, (b) u^*, (c) y^*, and (d) v^* of the centroid of the cylinders for the DKT process.

4.2 | Particle-Laden Channel Flow with an Obstacle

Particle-laden channel flow including an obstacle with collision and adhesion phenomena, which simulates the blood flow around the medical stent, was computed. The collision and adhesion effects were taken into account by our model and Tomita et al. [43], respectively. In the present computation, 200 moving cylindrical particles and an fixed obstacle were contained in the channel flow. The computational domain based on the particle diameter D was $0 \le x \le 100D$ and $0 \le y \le 40D$ with $4,000 \times 1,600$ grid points. The inflow and outflow boundaries were imposed at the $-x$ and $+x$ outer boundaries, respectively, and the no-slip boundary condition was imposed at the surfaces of the obstacle and cylindrical particles and the $-y$ and $+y$ outer boundaries, respectively. The grid size was set to be NPD $= 40$. The Reynolds number based on the inflow velocity, the viscosity coefficient, the fluid density, and the cylinder diameter was set to be 150. The density ratio of the fluid and small cylinders was set to be $\rho_d/\rho_f = 1.0$, and the height of the obstacle was $10D$. The initial velocities of the particles and the fluid were zero and unity, respectively. The initial positions of the particles were randomly chosen in the region of $5D \le x \le 95D$ and $2.5D \le y \le 37.5D$.

The vortex shedding from the obstacle occurs as shown in Figure 15. The flow behind the obstacle stagnates and particles adhere to the wall, and several particles are transported due to the vortices. Figure 16 shows the average particle velocity in the $x-$ and $y-$directions. The average velocity of particles is computed by every $\Delta y = 1.0D$ in the region of $x \ge 30D$. The particle velocity is initially zero, and particles are transported downstream due to fluid flow. As the flow field develops, the particles appear in the region near the bottom wall due to the recirculation region of the obstacle, and the average velocity of particles in this region is lower than that of other particles. In addition, the average particle velocity is zero at the lower wall due to adhesion (adhiered particles are shown in black in Figure 15). The velocity toward the wall increases due to the recirculation region of the obstacle so that adhesion occurs more frequently behind the obstacle as shown in Figure 16(B).

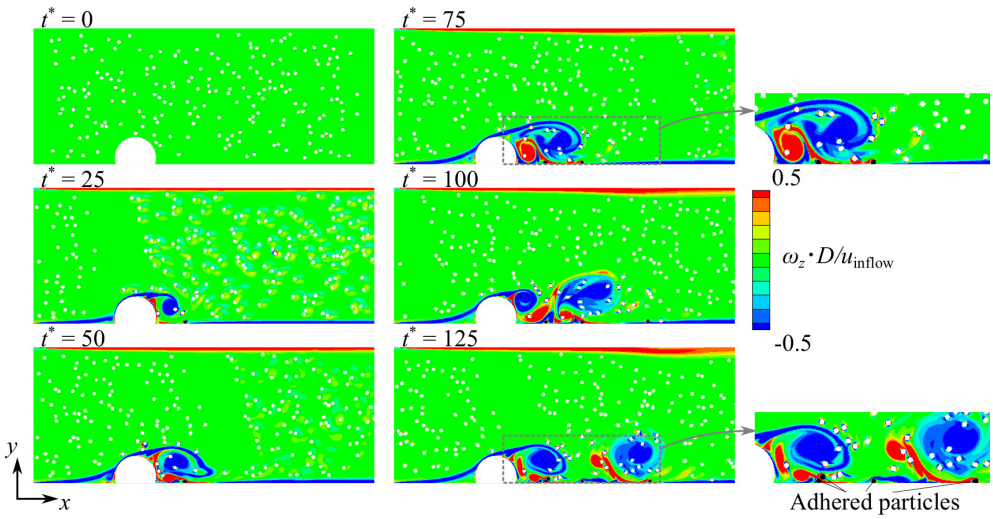

FIGURE 15 Vorticity distribution of particle-laden channel flow with an obstacle, including collision and adhesion effects.

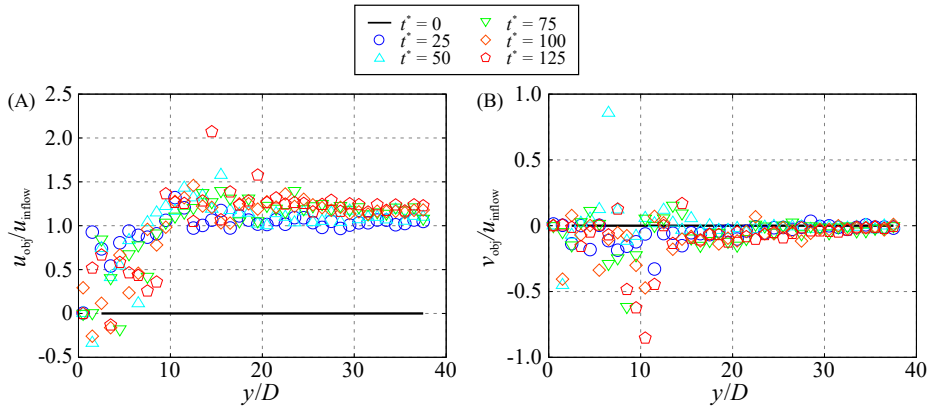

FIGURE 16 Average particle velocity where $x \geq 30D$: (A) x component and (B) y component.

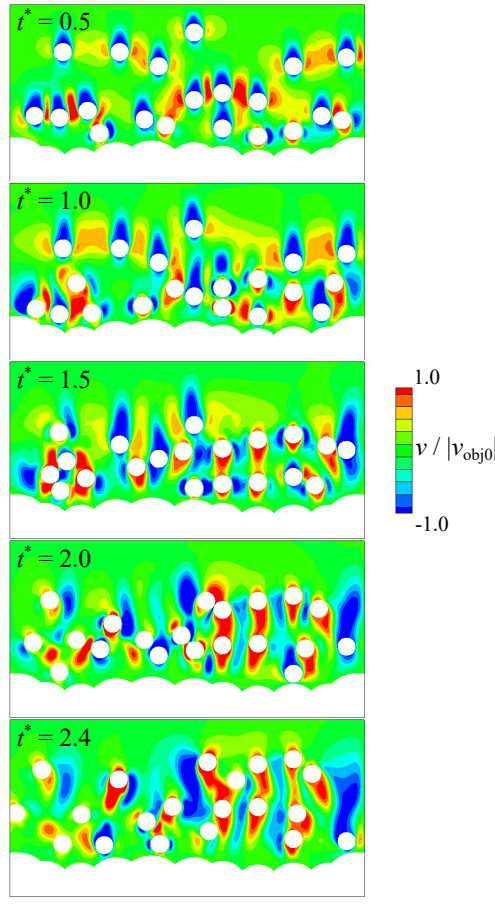

FIGURE 17 Velocity distribution of cylinder-wall interaction.

4.3 | Cylinder-Wavy Wall Interaction

Figure 17 shows snapshots of the interaction process between the cylinders and the wavy wall. The diameter of cylinders was D, and the size of the computational domain was $0 \leq x \leq 20D$ and $0 \leq y \leq 20D$ with $1,600 \times 800$ grid points. The grid resolution was set to be NPD $= 80$. The no-slip condition was imposed on the boundaries of the wavy wall and cylinders, and the outflow boundary condition was imposed at all outer boundaries. The coefficient of restitution of the cylinders and the wavy wall was set to be unity. The Reynolds number based on the fluid density, the kinematic viscosity coefficient, the diameter of the cylinder, and the magnitude of the initial cylinder velocity in the y-direction was set to be 50. Here, the initial cylinder velocities in x- and y-direction were $u_{\mathrm{obj0}} = 0$ and $v_{\mathrm{obj0}} = -1.0$, respectively. In this computation, the effect of the cylinders on the flow field was considered, but the effect of the fluid forces on the cylinder motion did not consider in order to discuss the total energy conservation of the cylinders. The 20 cylinders collide with and rebound from the wavy wall. Here, the number of the cylinder is less at a larger time because a part of cylinders goes out the computational domain after collided with the wavy wall. In order to avoid blowing-up of the computation, the quantities of the ghost nodes or outer boundaries are not updated when the out of the computational domain or ghost nodes are referred to determine the value of ghost nodes or outer boundary nodes.

The time history of the total energy of the cylinders and the collision frequency are shown in Figure 18. The kinetic total slightly changes during repeated cylinder-wall and cylinder-cylinder collisions. In this computation, the error in the total energy after the cylinder-wall and cylinder-cylinder interaction was approximately 6%. The kinetic energy loss occurs due to many time collisions, despite the small error for a single collision as discussed in Figures 9. In addition, the present algorithm assumes binary collision, but three or more body collisions occur in this computation, and thus further energy loss appears to be generated.

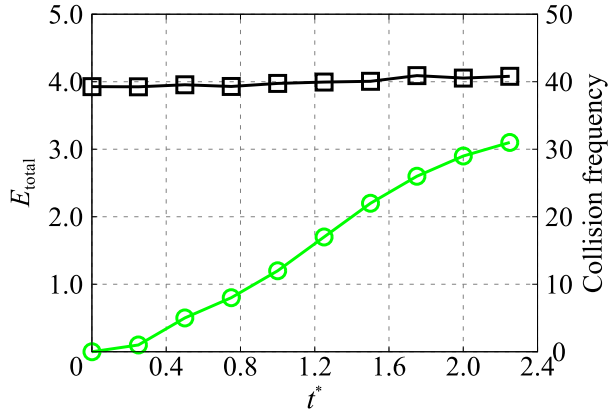

FIGURE 18 Time history of the total energy of the cylinders (black square symbols) and the collision frequency (green circular symbols).

5 | CONCLUSIONS

In the present study, we proposed a simple collision algorithm, which can be handled arbitrarily-shaped objects, for IBM flow solver. The proposed algorithm was implemented on the solid-liquid IBM flow solver based on the level set and ghost cell methods. This solver uses a particle-resolved approach by solving the Navier–Stokes and the Newton—Euler equations and can be account the collision and adhesion phenomena. The collision algorithm that we proposed in the

present study is a simple and efficient algorithm. It requires only little additional costs for the computation of collision phenomena because the collision detection and calculation are performed using the level set function and image point, which are incorporated into the original IBM flow solver. In addition, this algorithm is capable of solving the collision of arbitrarily-shaped objects.

The solver was firstly validated by solving flow over a fixed circular cylinder and sphere. For the two-dimensional case, the total, pressure, and viscous drag coefficients computed by the present solver showed good agreement with the previous numerical results. The Strouhal number of vortex shedding based on the oscillation of the lift coefficient and r.m.s. amplitude of the lift coefficient also showed good agreement with the previous numerical result. In addition, the grid convergence on these quantities was confirmed. Moreover, the collision algorithm was validated by calculating the cylinder-cylinder and cylinder-RBC interactions. In these tests, the total energy loss of objects due to the collisions process was investigated. For the cylinder-cylinder interaction, errors in the total energy of objects at NPD $= 10$ and 20 were approximately 2% and 0.63%, respectively. At finer grid of NPD $= 40$ and 80, the error decreases further, which confirmed grid convergence (error was approximately 0.2% at NPD $= 40$ and less than 0.01% at NPD $= 80$). The sufficient total energy conservation for the cylinder-RBC interaction was also confirmed similar to the cylinder-cylinder even though including rotational motion. In the three-dimensional case, the calculated drag coefficient showed good agreement with the standard drag curve and previous numerical results of the sphere, and grid convergence was also confirmed in each component of the drag coefficient. In addition, the sphere-sphere interaction was computed to investigate the grid and time step size sensitivity on the total energy conservation of objects in the three-dimensional case. As a result, the characteristics were similar to those of the two-dimensional cases. In addition, the sphere-wall interaction was computed and compared to the previous experimental results.

Several simulations with multiple moving objects and collision phenomena were conducted as demonstrations. As a first example, the DKT problem was computed. The time histories of the positions and velocities of the centroid of the cylinders in x- and y-directions were compared to previous numerical results. The present result was similar to the previous numerical results. As a second example, a particle-laden channel flow including an obstacle with collision and adhesion phenomena, which simulates the blood flow around the medical stent, was also computed. In this simulation, 200 cylindrical moving particles and an obstacle were contained in the channel flow. It was demonstrated that the present solver is capable of the simulation of the flow including collision and adhesion phenomenon of the particles. As a third example, the interaction between the cylinders and wavy wall were computed. It was confirmed that the computation can be conducted in stable, and the time history of the total energy was discussed.

The present results indicate the capability of solving a flow containing arbitrarily-shaped moving objects with collision and adhesion phenomena by a proposed simple algorithm.

ACKNOWLEDGEMENTS

The current computational resource was partially supported by the High Performance Computing Infrastructure (HPCI) hp160150 and hp170111, Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructure (JHPCN) jh180051-NAJ, and as collaborative research with Tohoku University Institute of Fluid Science (Grant J17I023, J18L023, J19L026). We appreciate their assistance and helpful support. The present study was supported by JSPS KAKENHI Grant Numbers 16K18018 and 18K03937.

ENDNOTES

REFERENCES

[1] Cundall PA, Strack OD. A discrete numerical model for granular assemblies. geotechnique 1979;29(1):47–65.

[2] Chorin AJ. Numerical solution of the Navier-Stokes equations. Mathematics of computation 1968;22(104):745–762.

[3] Tsuji Y, Tanaka T, Ishida T. Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe. Powder technology 1992;71(3):239–250.

[4] Tsuji Y, Kawaguchi T, Tanaka T. Discrete particle simulation of two-dimensional fluidized bed. Powder technology 1993;77(1):79–87.

[5] Udaykumar H, Shyy W, Rao M. Elafint: a mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries. International journal for numerical methods in fluids 1996;22(8):691–712.

[6] Ye T, Mittal R, Udaykumar H, Shyy W. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. Journal of computational physics 1999;156(2):209–240.

[7] Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. Journal of Computational Physics 2005;209(2):448–476.

[8] Ghias R, Mittal R, Dong H. A sharp interface immersed boundary method for compressible viscous flows. Journal of Computational Physics 2007;225(1):528–553.

[9] Takahashi S, Nonomura T, Fukuda K. A numerical scheme based on an immersed boundary method for compressible turbulent flows with shocks: application to two-dimensional flows around cylinders. Journal of Applied Mathematics 2014;2014.

[10] Luo K, Zhuang Z, Fan J, Haugen NEL. A ghost-cell immersed boundary method for simulations of heat transfer in compressible flows under different boundary conditions. International Journal of Heat and Mass Transfer 2016;92:708–717.

[11] Mittal R, Dong H, Bozkurttas M, Najjar F, Vargas A, Von Loebbecke A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. Journal of computational physics 2008;227(10):4825–4852.

[12] Feng ZG, Michaelides EE. The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. Journal of Computational Physics 2004;195(2):602–628.

[13] Lau EM, Huang WX, Xu CX. Progression of heavy plates from stable falling to tumbling flight. Journal of Fluid Mechanics 2018;850:1009–1031.

[14] Kim W, Lee I, Choi H. A weak-coupling immersed boundary method for fluid–structure interaction with low density ratio of solid to fluid. Journal of Computational Physics 2018;359:296–311.

[15] Xu L, Tian FB, Young J, Lai JC. A novel geometry-adaptive Cartesian grid based immersed boundary–lattice Boltzmann method for fluid–structure interactions at moderate and high Reynolds numbers. Journal of Computational Physics 2018;375:22–56.

[16] Schlanderer SC, Weymouth GD, Sandberg RD. The boundary data immersion method for compressible flows with application to aeroacoustics. Journal of computational Physics 2017;333:440–461.

[17] Riahi H, Meldi M, Favier J, Serre E, Goncalves E. A pressure-corrected Immersed Boundary Method for the numerical simulation of compressible flows. Journal of Computational Physics 2018;374:361–383.

[18] Lucci F, Ferrante A, Elghobashi S. Modulation of isotropic turbulence by particles of Taylor length-scale size. Journal of Fluid Mechanics 2010;650:5–55.

[19] HOSAKA M, NAGATA T, TAKAHASHI S, FUKUDA K. NUMERICAL SIMULATION ON SOLID–LIQUID MULTIPHASE FLOW INCLUDING COMPLEX-SHAPED OBJECTS WITH COLLISION AND ADHESION EFFECTS USING IMMERSED BOUNDARY METHOD. Multiphase Flow: Theory and Applications 2018;p. 167.

[20] Mehrabadi M, Murphy E, Subramaniam S. Development of a gas–solid drag law for clustered particles using particle-resolved direct numerical simulation. Chemical Engineering Science 2016;152:199–212.

[21] Zhang H, Trias FX, Gorobets A, Oliva A, Yang D, Tan Y, et al. Effect of collisions on the particle behavior in a turbulent square duct flow. Powder technology 2015;269:320–336.

[22] Mizuno Y, Takahashi S, Fukuda K, Obayashi S. Direct Numerical Simulation of Gas–Particle Flows with Particle–Wall Collisions Using the Immersed Boundary Method. Applied Sciences 2018;8(12):2387.

[23] Schneiders L, Günther C, Meinke M, Schröder W. An efficient conservative cut-cell method for rigid bodies interacting with viscous compressible flows. Journal of Computational Physics 2016;311:62–86.

[24] Das P, Sen O, Jacobs G, Udaykumar H. A sharp interface Cartesian grid method for viscous simulation of shocked particle-laden flows. International Journal of Computational Fluid Dynamics 2017;31(6-8):269–291.

[25] Hoomans B, Kuipers J, Briels WJ, van Swaaij WPM. Discrete particle simulation of bubble and slug formation in a two-dimensional gas-fluidised bed: a hard-sphere approach. Chemical Engineering Science 1996;51(1):99–118.

[26] Crowe CT, Schwarzkopf JD, Sommerfeld M, Tsuji Y. Multiphase flows with droplets and particles. CRC press; 2011.

[27] Kosinski P, Hoffmann AC. An extension of the hard-sphere particle–particle collision model to study agglomeration. Chemical Engineering Science 2010;65(10):3231–3239.

[28] Costa P, Boersma BJ, Westerweel J, Breugem WP. Collision model for fully resolved simulations of flows laden with finite-size particles. Physical Review E 2015;92(5):053012.

[29] Liu HR, Gao P, Ding H. Fluid–structure interaction involving dynamic wetting: 2D modeling and simulations. Journal of Computational Physics 2017;348:45–65.

[30] Glowinski R, Pan TW, Hesla TI, Joseph DD. A distributed Lagrange multiplier/fictitious domain method for particulate flows. International Journal of Multiphase Flow 1999;25(5):755–794.

[31] Glowinski R, Pan TW, Hesla TI, Joseph DD. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. Journal of Computational Physics 2001;169(2):363–426.

[32] Kempe T, Fröhlich J. An improved immersed boundary method with direct forcing for the simulation of particle laden flows. Journal of Computational Physics 2012;231(9):3663–3684.

[33] Favier J, Revell A, Pinelli A. A Lattice Boltzmann–Immersed Boundary method to simulate the fluid interaction with moving and slender flexible objects. Journal of Computational Physics 2014;261:145–161.

[34] Ardekani MN, Costa P, Breugem WP, Brandt L. Numerical study of the sedimentation of spheroidal particles. International Journal of Multiphase Flow 2016;87:16–34.

[35] Munjiza A, Andrews K. NBS contact detection algorithm for bodies of similar size. International Journal for Numerical Methods in Engineering 1998;43(1):131–149.

[36] Schinner A. Fast algorithms for the simulation of polygonal particles. Granular Matter 1999;2(1):35–43.

[37] Yao Z, Wang JS, Liu GR, Cheng M. Improved neighbor list algorithm in molecular simulations using cell decomposition and data sorting method. Computer physics communications 2004;161(1-2):27–35.

[38] Zhang P, Galindo-Torres S, Tang H, Jin G, Scheuermann A, Li L. An efficient Discrete Element Lattice Boltzmann model for simulation of particle-fluid, particle-particle interactions. Computers & Fluids 2017;147:63–71.

[39] Kim J, Moin P. Application of a fractional-step method to incompressible Navier-Stokes equations. Journal of computational physics 1985;59(2):308–323.

[40] Young DM. Iterative solution of large linear systems. Elsevier; 2014.

[41] Nonomura T, Onishi J. A Comparative Study on Evaluation Methods of Fluid Forces on Cartesian Grids. Mathematical Problems in Engineering 2017;2017.

[42] Baraff D. An introduction to physically based modeling: rigid body simulation II—nonpenetration constraints. SIGGRAPH course notes 1997;p. D31–D68.

[43] Tomita A, Tamura N, Nanazawa Y, Shiozaki S, Goto S. Development of virtual platelets implementing the functions of three platelet membrane proteins with different adhesive characteristics. Journal of atherosclerosis and thrombosis 2014;p. 26203.

[44] Zhang HQ, Fey U, Noack BR, König M, Eckelmann H. On the transition of the cylinder wake. Physics of Fluids 1995;7(4):779–794.

[45] Clift R, Gauvin W. Motion of entrained particles in gas streams. The Canadian Journal of Chemical Engineering 1971;49(4):439–448.

[46] Johnson T, Patel V. Flow past a sphere up to a Reynolds number of 300. Journal of Fluid Mechanics 1999;378:19–70.

[47] Luo K, Wang Z, Fan J. A modified immersed boundary method for simulations of fluid–particle interactions. Computer methods in applied mechanics and engineering 2007;197(1-4):36–46.

[48] Nagata T, Nonomura T, Takahashi S, Mizuno Y, Fukuda K. Investigation on subsonic to supersonic flow around a sphere at low Reynolds number of between 50 and 300 by direct numerical simulation. Physics of Fluids 2016;28(5):056101.

[49] Chee C, Lee H, Lu C. Using 3D fluid–structure interaction model to analyse the biomechanical properties of erythrocyte. Physics Letters A 2008;372(9):1357–1362.

[50] Thompson MC, Leweke T, Hourigan K. Sphere–wall collisions: vortex dynamics and stability. Journal of Fluid Mechanics 2007;575:121–148.

[51] Fortes AF, Joseph DD, Lundgren TS. Nonlinear mechanics of fluidization of beds of spherical particles. Journal of Fluid Mechanics 1987;177:467–483.

[52] Koblitz A, Lovett S, Nikiforakis N, Henshaw WD. Direct numerical simulation of particulate flows with an overset grid method. Journal of Computational Physics 2017;343:414–431.